Real-Time Multi-Scale Object Detection in Surveillance Using Hybrid Transformer Architecture

Roshan D Suvaris[1], Rahul Suryodai[2], S. Narayanasamy[3], Dr. Aanandha Saravanan[4], Raman Kumar[5], Dr. P N V Syamala Rao M[6], Elangovan Muniyandy[7]

Asst. Professor, Nitte (Deemed to be University), NMAM Institute of Technology (NMAMIT), Nitte, India [1]
Senior Data Engineer (Data Governance, Data Analytics: Enterprise Performance Management, AI&ML), USA[2]
Department of Computer Science and Engineering, J. J. College of Engineering and Technology, Tiruchirappalli, Tamilnadu, India[3]
Professor, Department of ECE, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India[4]
University School of Mechanical Engineering, Rayat Bahra University, Mohali, India[5a]
Faculty of Engineering, Sohar University, Sohar, Oman[5b]
Assistant Professor, Department of CSE, SRM University AP, Amaravati, Andhra Pradesh, India – 522240[6]
Department of Biosciences-Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai - 602 105, India[7a]
Applied Science Research Center, Applied Science Private University, Amman, Jordan[7b]

***Abstract*—Real-time surveillance systems require accurate and efficient object detection to ensure safety and situational awareness. Existing methods, such as YOLOv5 and Vision Transformer-based detectors, often struggle to reliably identify small, distant, or occluded objects while maintaining real-time inference, limiting their applicability in complex surveillance environments. To address these challenges, this study proposes PRISM, a hybrid Transformer–YOLOv8 framework that integrates fast local feature extraction with global contextual refinement. The method introduces two novel components: i) a Context-Aware Feed Forward Network (CA-FFN) within the Vision Transformer (ViT), which dynamically weights channel features to reduce redundancy and enhance global context modeling, and ii) Cross-Scale Attention Skip Connections (CSASC) for selective fusion of multi-scale YOLOv8 and ViT features, improving detection of small or occluded objects. The model is implemented in PyTorch and trained on a comprehensive surveillance dataset consisting of pedestrians, vehicles, bicycles, bags, and miscellaneous objects. Experimental evaluation demonstrates that PRISM achieves 96% accuracy, a significant improvement of ~4–5% over baseline methods, with robust performance across all object categories. Key performance indicators verify the reliability of the model to real-time usage, and the lightweight design makes it edge deployable. These findings imply that PRISM can be used to provide a speed-accuracy balance in a complex and dynamic setting, which is more efficient than the current methods. The study also notes the partial extensions, such as the incorporation of multi-sensors and continuous video streams to do time modeling as an extension, which will offer a good base to the next-generation intelligent surveillance systems.**

***Keywords*—Real-time object detection; hybrid transformer–YOLOv8; Context-Aware Feed Forward Network (CA-FFN); Cross-Scale Attention Skip Connections (CSASC); surveillance video analytics; multi-scale feature fusion**

I. Introduction

Real-time surveillance has turned into an inseparable component of the contemporary governance of the city, social security and management of vital infrastructure [1]. Since the idea of smart cities has entered a new phase and the necessity of automated monitoring devices [2] is growing, the ability to properly distinguish and classify objects in dynamic and complicated environments has received primary importance [3]. These traditional computer vision algorithms have been largely reliant on handcrafted features that are sensitive to varying light conditions, occlusion, and size, and hence limit their application in real-world surveillance [4]. With the emergence of deep learning (DL) algorithms, and in particular convolutional neural networks (CNNs) [5], object detection has gotten a significant boost due to automated feature extraction and strong recognition features. The You Only Look Once (YOLO) family of CNN-based [6] detection methods has become one of the most popular frameworks in real-time detection because it operates in a single stage, and all images are processed in a single pass, having an impressive inference speed [7]. However, the YOLO-based models are mostly centered on the local spatial information, disregarding the contextual relationships at a large scale, which may worsen the performance of the model on the detection of small, hidden, or distant objects [8]. ViTs, on the other hand, make use of self-attention mechanisms to learn global dependencies [9], which give a better representation of more complex scenes and greater detection of difficult-to-detect object classes. However, ViTs are computationally expensive and latency-prone, which makes them less ideal for real-time execution on edge devices [10]. Hybrid models using CNN backbones and transformers have also been studied, but most existing methods are inclined to excessive computational cost or non-optimal multi-scale feature integration, which leads to a trade-off between efficiency and accuracy [11], [12].

To address these challenges, this study presents PRISM, a new hybrid system of scalable real-time monitoring of surveillance objects. PRISM combines the property of extracting features of objects within a short time frame used in YOLOv8 with a lightweight ViT module, with an addition of a Context-Aware Feed-Forward Network (CA-FFN) that boosts the network's understanding of the objects in their entirety. This is done using a Cross-Scale Attention Skip Connection (CSASC) mechanism that is effective in producing local and global features at multiple scales without compromising speed and accuracy. The framework integrates powerful preprocessing, data augmentation, and CIoU-based optimization to achieve maximum performance in the detection of various categories of objects such as pedestrians, vehicles, bicycles, and miscellaneous objects. PRISM aims to work around the shortcomings of the current method, combining the strengths of YOLOv8 and ViTs to provide a better detection of small, occluded, and distant objects, preserving the performance of real-time detection. The proposed framework would be applicable to both edge and cloud deployments and would be flexible, scaled, and robust in the present-day surveillance applications.

A. Research Motivation

The increased complexity of urban observation contexts (i.e., occlusion, changing light, and object size variability) requires detection systems to achieve high accuracy while meeting real-time demands [13]. Existing models rely on speed or are not generalizable when faced with challenging situations, which creates the need to develop a generalizable, scalable hybrid approach to robust object recognition.

B. Significance of the Study

This study addresses important gaps in present-day surveillance object detection by utilizing multi-scale contextual learning and fast inference. The proposed framework improves the detection of small, occluded, and distant objects to collect reliable data for a better and more effective public safety, traffic management, and infrastructure protection. Lastly, this solution is deployable across edge and cloud devices.

C. Recent Innovation and Challenges

Recent advancements, such as transformer-based models and hybrid CNN-transformers, have increased the quality of representation of global features. However, there are still difficulties in achieving a balance between computational efficiency, multi-scale feature fusion, and real-time operation. Integrating lightweight attention mechanisms effectively is still a primary challenge to implement practical, scalable surveillance solutions in changing, dynamic environments.

D. Key Contributions

- Hybrid Framework Design: Development of PRISM, a scalable object detection framework that combines fast local feature extraction with global contextual refinement for robust real-time surveillance.
- Novel Fusion Mechanism: Innovation of CSASC to improve multi-scale feature fusion while ensuring computational efficiency.
- Transformer Optimization: Addition of a CA-FFN to the lightweight ViT module to enhance global feature learning on small, occluded, and faraway objects.
- Robust Preprocessing Pipeline: Use of a systematic data cleaning, augmentation, normalization, and adaptive training approach to achieve model generalization in various surveillance environments.

The remaining section of this study will follow the below organization. In Section II, prior research in real-time detection in the transfer application will be explored, examining findings and restrictions of YOLO and Transformer-based models. In Section III, the most notable challenges in surveillance object detection are identified. These are small object recognition, occlusions, and computational efficiency. The proposed Scalable Transformer-YOLO Model methodology is discussed in Section IV and includes preprocessing of the dataset, architecture of the model, and training methodologies. Section V presents results from experimentation, performance metrics, and comparisons to other conventional models. Section VI concludes with a few conclusions, along with some major discussion and potential future improvement work for the real-time surveillance detection system outcomes.

II. Related Works

Ouyang [14] was proposed as a hybrid system, which combines the concepts of DETR and YOLO to become more robust and more accurate. It is a two-stage pipeline architecture in which the bounding boxes of objects were predicted by YOLO extremely quickly. Then, to add high precision transformer-based model refinement was employed. On the COCO dataset, DEYO has achieved an achievement of 52.1 AP and has surpassed the traditional YOLO models in both detection and performance. The combination of YOLO and transformers assists in the local and global contexts, which progresses the object recognition in difficult scenes. Although DEYO improved the accuracy, it had a negative response as it took a long time to converge and was computationally too costly to use in real-time. The principal disadvantage was its accompanying downside of preparing a large amount of training data and adjusting hyperparameters, consuming resources. Additionally, the transformer-based refinement added even more latencies, restricting its capacity for real-time surveillance applications.

Song et al. [15] proposed ViDT, a novel object detection model based solely on the ViTs, which has subsequently improved detection accuracy. ViDT, in contrast to earlier CNN-based models, fabricates a mechanism for retrieving long-range image dependencies, which has improved object recognition. The model has displayed an AP of 49.2 on the COCO dataset; thereby, it outperforms conventional object detection models. The capacity to learn contextual relationships was one of the main benefits of this model and proved useful in situations when objects were interacting. It is, however, a resource-consumptive and, therefore, not applicable to near-real-time edge solutions that require improved resource utilization. The other weakness is that it is based on training on large datasets, and this usually translates to the transformers requiring much more labeled data in order to perform optimally. The necessity to have lightweight models of transformers that would create a

reasonable balance between accuracy and efficiency in computation has been mentioned in the work.

Wang et al. [16] suggested Mamba YOLO, a state-of-the-art model for object detection, which incorporated the use of SSMs in the YOLO to enhance efficiency and accuracy. Mamba YOLO, which makes use of SSMs in the continuity of its one-step thinking process, allows tracking of objects through time. In this way, it greatly aids by reducing the complexity of computation and hence completely enables the applications of Real-time in autonomous systems and surveillance. The model demonstrated huge progress on the COCO and VOC datasets with large improvements compared to classical YOLO versions on both precision and recall. Another weakness was the inability to perform better in extreme light conditions because objects with poor contrast were hard to notice. Nonetheless, Mamba YOLO had potential for making a valuable contribution to the real-time detection area, given the blend of YOLO's speed with SSMs' sequential page processing abilities.

G. and B. [17] created YoloTransformer-TransDetect, a hybrid in that their framework merges YOLO with transformer-based attention mechanisms to implement defect detection. The sequence included first using YOLO to quickly identify defects, followed by the transformer module that worked to extract features at a finer level of grain to improve accuracy. There was better performance of the model in detecting defects in steel tubes, as well as an increase in the detection rates. The technique was not able to work with extremely complicated or visually similar defect patterns in a single pass and had to undergo additional post-processing. A tiny defect and occlusion, one of the common problems in industry, was classically privileged when the method could reliably, if not efficiently, identify. The research found that some feature learning and localization were realized using transformer-based techniques along with YOLO. Its operation efficiency, however, was questioned, and restricted its use in real-time. The future research undertakes the aim of optimization of transformer block to offer convenience in shortcomings.

Li, Yan, and Shi [18] introduced PP-YOLOE, a new version of YOLO for object detection, which comes with a multi-scale attention mechanism. The adaptive scaling method improves feature extraction to better specify fine distinction details in every scale of any object. Application of these mechanisms significantly enhanced the total accuracy of detection on the COCO data set, mainly on small objects, thus are highly overlooked in previous traditional YOLO models. This network, with the help of attention-based techniques, was rather a skilled object localizer in cluttered and occluded scenes, where it provided fewer false positives. Latency can limit applications in which real-time decision-making is required as video surveillance or even autonomous driving. It is also suggested in the study that lightweight models of transformer should be followed up in order to achieve a good speed-accuracy trade-off.

Su et al. [19] came up with a different model of breast mass detection as they divided the mass into LOGO and applied the strengths of YOLOv5 to it. The algorithm is a two-step process and involves both segmentation and detection of breast masses. First, the image is segmented and cropped with the help of a YOLOv5L6 model on high-resolution mammograms; then, the image and the area with breast masses are processed by various global and local transformer branches; finally, they are combined to yield the final segmentation output. Although these are encouraging figures, integration of transformers has computational difficulties, and it is not an easy task to implement it in a regular hospital system that has low computing capabilities. The study revealed that fusing transformers with YOLO improves boundary refinement and lesion detection but requires excessive data augmentation for generalization onto other imaging datasets.

Shah and Tembhurne [20] developed Defect Transformer, a hybrid architecture utilizing transformers, set to leverage surface defects in industrial applications, which is based on attention-based feature extraction for localization and classification of defects, as well as other aspects. The model was reported to be more accurate in detecting microscopic defects compared to the traditional CNN models. It is however, limited in application when it comes to real-time monitoring due to extremely high levels of computational complexity. This study demonstrated that perceiving the world context can be useful in bettering the process of defect classification. Although it can have multiple applications, there are major limitations to its implementation in real-time applications because of the memory and processing requirements of embedded systems.

Shang et al. [21] presented the Defect-Aware Transformer Network, a deep learning framework designed for intelligent defect detection in modern industrial applications. The model integrated a self-attention mechanism to improve feature representation, which helps to detect those defects that were considered subtle and fine-grained, as is often missed with traditional convolutional models. This was in addition to the mix that was the transformer-based feature extraction and the conventional DL methodologies, which offered good learning of both local and global defect patterns. The experimental tests have demonstrated that the model resulted in a significant improvement of the trade-off between the accuracy and robustness of the classification, in particular, against defects with low contrast and textured backgrounds. In spite of these strengths, the authors also mentioned that their method is computationally expensive and thus not practical in real-time use in the automation of industry.

Current object detection studies highlight hybrid models, where YOLO is combined with transformers to make a balance between speed and precision. Hybrid methods enhance accuracy, but are computationally expensive. Pure transformer-based models detect long-range dependencies and contextual associations, but depend on large datasets and intensive resources. Some models combine sequential modules or multi-scale attention to achieve real-time efficiency and identify fine details. Although refinements based on transformers improve overall detection performance and localization, they also increase latency and computational requirements, which reduces their feasibility in real-world applications such as surveillance, autonomous systems, and industrial defect detection.

III. Problem Statement

Real-time object detection forms are one of the core components of surveillance systems where accuracy, flexibility, and efficiency are of equal significance. YOLO-based models are appreciated due to their lightweight architecture and fast inference that allows them to be deployed in edge cases and real-time systems. However, they do not properly identify small, distant, or hidden objects that usually appear in crowded and dynamic surveillance conditions [16]. Transformer-based models, on the other hand, are good at capturing global spatial context and enhance the quality of detection, but are costly to compute, and thus are restricted in their use in resource limited or real-time environments [22]. Hybrid architectures that integrate YOLO with transformer-based architectures have been reached to overcome those drawbacks, but they are generally accompanied by additional issues such as slow convergence, large memory complexity, and scalability constraints in the field environment. These drawbacks leave a research gap that hinders the development of smart surveillance. This is dealt with in the present work by the development of an optimized hybrid framework that trades accuracy, contextual learning, and efficiency of inference both in edge and cloud-based environments.

IV. Proposed PRISM Framework for Real-Time Object Detection

The suggested PRISM framework is modelled as the powerful and scalable object-detecting built-in surveillance system in real-time that is able to take into account the challenges of crowded scenes, changing illumination, partiality, and small sizes of items. The architecture is a hybrid between the convolutional and transformer-based architecture to trade local detail speech and global contextual reasoning. YOLOv8 is essentially the backbone network, and is effective at extracting multi-scale local features giving spatial precision and fast inference, which are essential to real time implementation. Following the extraction step, a lightweight ViT fine-tunes them by capturing long-range dependencies and high-level semantic relationships across the scene. The ViT component is enhanced with a CA-FFN to perform efficient channel-wise recalibration of the features as well as minimizing redundancy in the output. In order to fill the gap between local spatial information and global context, a new CSASC mechanism is added for enabling the proper merging of YOLOv8 features with ViT-refined results. The combined detection head subsequently executes bounding box regression and classification, which is optimized using CIoU-based loss and adaptive learning policies. Overall, this hybrid architecture provides a high-accuracy real-time system that is still computationally lightweight and well-suited for cloud and edge deployments in surveillance systems.

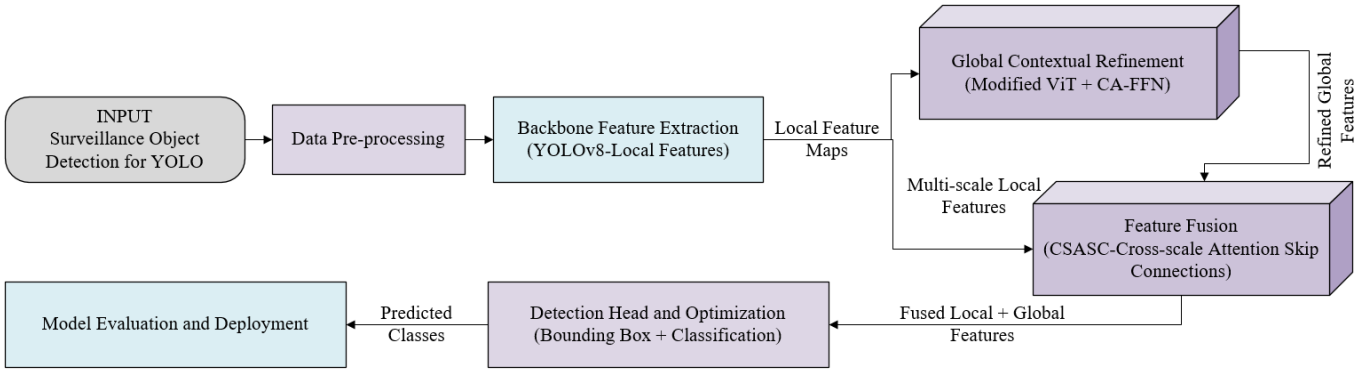

Fig. 1. Proposed PRISM framework.

Fig. 1 shows the process used to detect by the proposed system. The preprocessing of input video streams is followed by feature extraction by means of the YOLOv8 backbone and a lightweight ViT. The extracted features are then used to perform object detection and classification after which the outputs undergo post-processing to give a higher caliber. Evidence is formed to evaluate the quality of detection. The framework is then capable of implementing real-time decision-making or deployment on edge and cloud platforms, which guarantees scalability, efficiency, and flexibility in various settings of surveillance

A. Dataset Overview

The "Surveillance Object Detection Dataset for YOLO" [23] from Kaggle aims to create a training and test dataset with labeled training images specifically for surveillance real-time object detection applications. Approximately, 1,000 annotated images were taken based on real-world surveillance footage, and they depict vehicles and pedestrians, as well as objects of interest, the images are annotated with bounding boxes in the YOLO format that can be used with YOLOv8 models. In every annotation file, there are the class IDs, and coordinates of the bounding box in normalized form (0 to 1) which implies that it is resolution-independent. The dataset is most suitable in the real-time security applications including smart monitoring, anomaly detection, and traffic monitoring whereby strong object detection is a must. It has a great foundation with its formal annotation format and core object types that offer a strong and realistic platform to train YOLOv8 based surveillance models with better accuracy and reliability. The dataset parameters were given in Table I.

TABLE I. DATASET ATTRIBUTES

Attribute	Description	Attribute
Total Images	~1,000 images (varies based on dataset version)	Total Images
Image Resolution	Varies; normalized to 640×640 pixels	Image Resolution
Object Classes	Vehicles, pedestrians, suspicious objects	Object Classes
Annotation Format	YOLO format (TXT files with bounding box details)	Annotation Format
Data Source	Surveillance camera feeds	Data Source

B. Dataset Preprocessing

The input data required for the YOLOv8 model must be acquired using preprocessing techniques for higher precision, generalization, and stability in real-world applications of surveillance. The preprocessing techniques include data cleaning, conversion of format, data augmentation, normalization, and data splitting. These are all important in improving the dataset and the performance of the model.

1) Data cleaning: Cleaning data involves finding and fixing errors, inconsistencies, gaps, or duplicate entries in a dataset. It is important for verifying the accuracy, consistency, and reliability, all of which are required in order to construct effective and trustworthy machine learning models.

a) Identification and removal of corrupt and damaged images: Images that have been corrupted by the files, have an encoding problem, or have been incompletely downloaded are filtered out of the dataset before training [24]. Corrupt files lead to loading errors, higher overhead of computation, or non-batch processing, and therefore instability during model training. The images in the dataset that cannot be read might cause the model to drop out some training examples, and this can also lead to inconsistency in the learning of features and disrupt gradient changes.

Corrupt images are those that fail the image integrity checks. It ensures that all files are complete, well-formatted, and can be accessed. Such images should be deleted, particularly for programs, such as DL models like YOLOv8, where model performance can degrade due to missing or incomplete data. Suppose I_c refers to the number of corrupt images, and I_t to the total number of images in the dataset. The percentage of corrupt image removal can be calculated as in Eq. (1):

$$P_c = \left(\frac{I_c}{I_t}\right) \times 100 \quad (1)$$

b) Filtering mislabelled or empty annotations: Every image in the dataset should ideally have a YOLO annotation file containing bounding box coordinates and class labels. If an annotation file is missing, empty, or incorrectly labeled, the affected images will be adjusted or removed accordingly. Mislabelled annotations could lead to incorrect detections, misclassifications, and an increase in the model's false positive rate, thereby lowering overall detection accuracy.

c) Identification and elimination of duplicate images: Duplicate images lead to bias in the dataset, whereby the model overfits to the frequently seen samples. It follows when the model picks redundant sonic patterns and reduces its generalization ability to unseen test data of the surveillance system. Duplicates are mostly generated if datasets were joined, for example, in cases of automatic image gathering into surveillance systems, or data augmentation errors.

As in Table II, the removal of these problematic images makes way for a better dataset, thereby leading to improved model efficacy, lesser false detection, and improved generalizations in real surveillance environments. This refined dataset is now improved and made apt for preprocessing, augmenting, and training for the YOLOv8 model.

TABLE II. DATA CLEANING OPERATIONS

Issue Identified	Issue Identified	Percentage of Dataset (%)
Corrupt/Damaged Images	120	2.40%
Mislabeled/Empty Annotations	340	6.80%
Duplicate Images	215	4.30%
Total Removed	675	13.50%

2) Format conversion: To ensure conformity, all the images are also transformed into JPEG (.jpg) or PNG (.png) file format. The bounding box annotations are also YOLO format, representing the objects by normalized coordinates in the image frame. The normalizing equation of the bounding box is in Eq. (2):

$$\mathrm{x}' = \frac{x}{W}, \mathrm{y}' = \frac{y}{H}, \mathbf{w}' = \frac{w}{W}, \mathrm{h}' = \frac{h}{H} \quad (2)$$

To make the model more robust and diverse in terms of data, augmentation is done. These changes enable the model to train to identify items in many different situations of low light, motion blur and occlusions.

3) Data augmentation: In order to increase the dataset diversity and robustness of the model, augmentation is applied. Such transformations allow the model to learn to recognize objects in various conditions ranging from low light, motion blur, and occlusions.

The augmentations are used randomly to prevent overfitting, thereby, preserving model generalization in real-world surveillance environments, as provided in Table III.

TABLE III. AUGMENTATION TECHNIQUES AND THEIR EFFECTS

Augmentation Technique	Effect on Model Performance
Scaling (Zoom in/out)	Helps recognize objects at varying distances
Rotation (±15°)	Improves robustness to different camera angles
Flipping (Horizontal)	Enhances ability to detect objects in flipped views
Contrast Adjustment	Compensates for poor lighting conditions
Gaussian Noise Addition	Improves robustness to real-world noise

4) Normalization: This process of normalizing pixel intensities is highly significant as preprocessing and allows making the training of the model stable and efficient. The 0 to 255 raw pixel values in DL typically cause certain wild variation in the weight update in back propagation. These variations can lead to the unstableness of gradient descent, sluggish convergence, and model optimization. In order to solve these issues, pixel intensity is modified, and the intensity values are scaled within a fixed range between 0 and 1 in order to ensure a uniform intensity distribution across all images for the model to learn.

This enables the model to learn features more robustly by diminishing variances in brightness and contrast across the various images. It also allows for faster convergence that will help mitigate the effects of vanishing and exploding gradients. Generally, the normalization can be mathematically expressed, as in Eq. (3):

$$\mathrm{I}' = \frac{\mathrm{I} - I_{min}}{I_{max} - I_{miin}} \tag{3}$$

where, the original pixel value is denoted by I, while I_{min} and I_{miin} represent the minimum and maximum pixel intensities, typically 0 and 255, respectively. The normalized pixel value, I', is scaled within the range [0, 1]. This normalization ensures that the input distributions across all images are consistent, enabling the model to train more effectively. As a result, the model achieves better generalization, leading to improved accuracy in real-time object detection tasks.

5) Dataset splitting: As soon as preprocessing is done, the dataset is separated into orderly and effective subsets: training, validation, and testing for testing model performance in a balanced way. A balanced separation of the dataset is very crucial to avoid overfitting, optimized the learning of the model, and accurate evaluation of unseen data. The above dataset is split with reference to the standard splitting into 70% of the entire data for training, and allows learning the shapes, object features, and positions of the bounding box,20% for validation to ensure hyperparameter tuning and performance monitoring during the training process. The validation dataset helps in ensuring that the model generalizes well while protecting against overfitting onto the training dataset.10% set aside for testing to give unbiased feedback about the model's detection accuracy on data it has not seen. The details of the distribution of this dataset into different test sets are shown in Table IV.

TABLE IV. DATASET PARTITIONING

Dataset Partition	Number of Images	Percentage (%)
Training Set	700	70%
Validation Set	200	20%
Test Set	100	10%

C. Feature Extraction

The YOLOv8 is used in the PRISM framework as the base of rapid local feature detection in the frames of surveillance video. The processing of every frame is done separately by convolutional layers of the YOLOv8 that identify low- and mid-level spatial features, including edges, textures, and contours of objects. These characteristics are essential in the detection of small, remote or partially obscured objects that are major problems in real-time video surveillance images. The architecture of YOLOv8 produces multi-scale feature maps, with multi-resolution detection heads which find objects proficiently. This feature enables precise identification of large, clearly defined objects, including vehicles, and smaller, less obvious objects, such as bicyclists or pedestrians. Formally, the extraction at each layer is given as Eq. (4):

$$F_l = \sigma(W_l * F_{l-1} + b_l) \tag{4}$$

where, F_l is feature map at layer l, W_l and b_l are convolutional kernel and bias, $*$ is convolution, and σ is the activation function. This allows YOLOv8 to obtain spatial and contextual information in every frame of the video due to the hierarchical feature map aggregation. The heads of detection make predictions of bounding boxes and class likelihoods at scales of different sizes at detection time, where larger objects are more sensitive.

YOLOv8 is a real-time, high-throughput spatial fidelity sequentially processing in the case of video datasets. The learned feature maps are subsequently fed to the small ViT, which learns the long-range dependencies and the intrinsic frame contextual relations. Such a mix is what guarantees that PRISM can guarantee the good performance of detecting the dynamic scenes and can handle the application of support to the use of occlusions, motion blur, and varying object sizes without reducing its real-time performance.

D. Global Contextual Refinement

After the local feature extraction stage, which is the YOLOv8, the PRISM framework adopts a lightweight ViT to refine the global context, i.e., long-range dependencies across the extracted robustly feature maps. Unlike the existing ViTs, the proposed architecture presents a novel addition to the traditional FFN layer with the CA-FFN that offers a higher efficiency level, less redundancy of features, and better edge correctness. The modified ViT framework is given in Fig. 2.

The CA-FFN functions by initially performing a regular linear transformation on the input feature $X \in \mathbb{R}^{N \times D}$, where N represents the number of tokens extracted from the YOLOv8 feature map, and D is the dimension for embeddings. Rather than a simple MLP, the transformed feature is subject to channel-wise attention, selectively amplifying informative features while suppressing irrelevant or redundant patterns:

$$A = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right), \quad X' = AV \tag{5}$$

In Eq. (5), Q, K, V are the query, key, and value matrices from X, and d_k is the scaling factor. The attention output $X'$is next fed into dynamic feature gating, where a learnable gating vector $G \in \mathbb{R}^D$ is used to control the flow of information, given in Eq. (6):

$$X_{CA-FFN} = G \odot GELU(X'W_1 + b_1)W_2 + b_2 \tag{6}$$

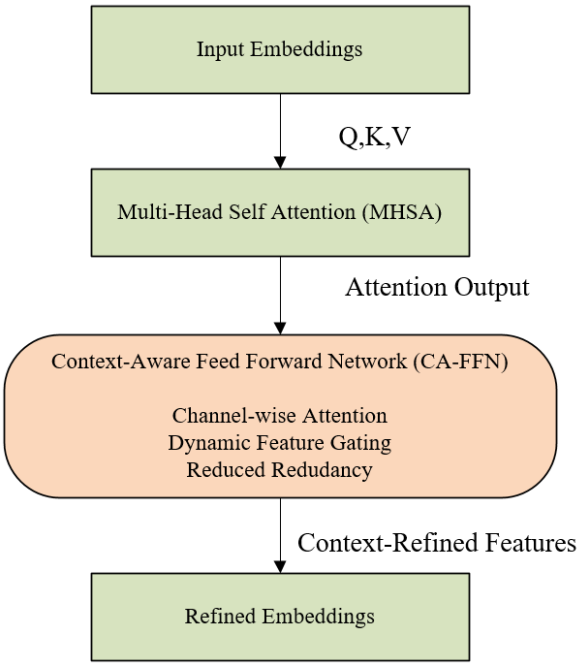

Fig. 2. Global contextual refinement.

where, W_1, W_2 and b_1, b_2 are the trainable weights and biases of the FFN, $\odot$ represents element-wise multiplication, and GELU applies non-linear activation. The gating mechanism enables the network to suppress less useful channels and concentrate computational resources on important features, leading to a significant improvement in efficiency, particularly for edge deployment applications with limited resources.

The PRISM framework, through the inclusion of CA-FFN, makes sure that global contextual information is narrowed without over-inflating computational complexity. The novelty can be easily stated: instead of the traditional ViTs, the FFN is substituted by the context-aware, attention-driven, gated module, which is, at the same time, more effective in terms of feature representation, redundancy, and lightweight architecture to fit into real-time video surveillance. The given step is essential to make sure that the fused YOLOv8 features have contextualized information, increasing the ability to find small, obscured, or visually challenging objects without making real-time performance worse.

E. Feature Fusion Mechanism

The PRISM framework uses representations by taking local feature representations provided by YOLOv8 and contextually enriched global feature representations provided by the CA-FFN-enhanced ViT, integrating them with a Feature Fusion Mechanism. Conventional skip-connections only concatenate, or add, feature maps across layers, which can tend to dilute salient information and cannot prioritize important multi-scale cues that are necessary to detect small, distant, or occluded objects. In order to address this limitation, the proposed framework includes a Cross-Scale Attention Skip Connection (CSASC), a new selective integration feature; its architecture is given in Fig. 3.

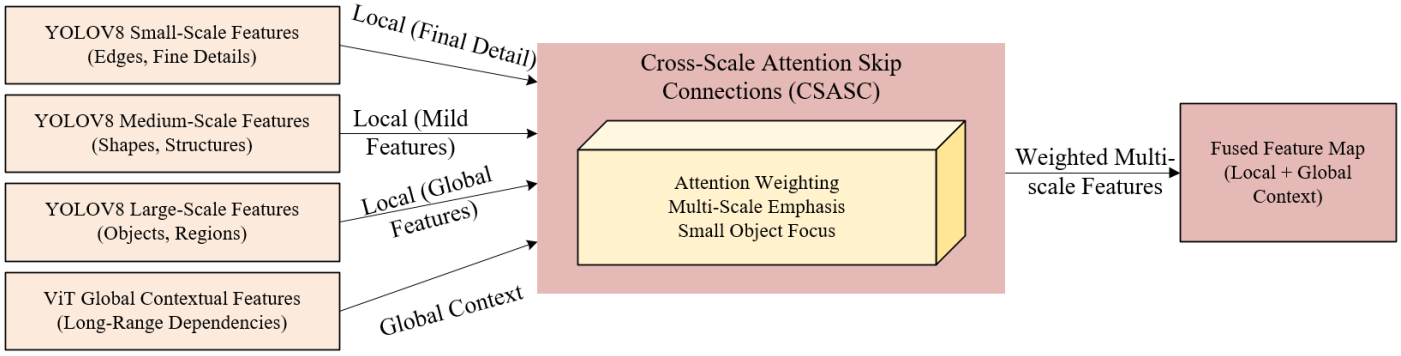

Fig. 3. CSASC fusion mechanism.

The CSASC functions by initially aligning the spatial sizes of YOLOv8 feature maps $F_{YOLO} \in \mathbb{R}^{H_1 \times W_1 \times C_1}$ and refiner-ViT features $F_{ViT} \in \mathbb{R}^{H_2 \times W_2 \times C_2}$ through bilinear interpolation and token reshaping. Then, attention maps for every scale are calculated to weigh the significance of various resolution features, as given in Eq. (7):

$$\alpha = Softmax\big(Conv_{1\times1}(F_{YOLO})\big) + Conv_{1\times1}(F_{ViT}) \quad (7)$$

$$F_{fused} = \alpha \odot F_{YOLO} + (1-\alpha) \odot F_{ViT} \quad (8)$$

In Eq. (8), α is the attention-weighting tensor derived from attention, $\odot$ is element-wise multiplication, and $Conv_{1\times1}$ is used to reduce channel dimensions for computational efficiency. This attention-based fusion allows for more attention to features with larger contexts, such as small objects, far-away vehicles, or partially occluded pedestrians, and less attention to regions that do not contain informative features.

With the application of CSASC, the framework rewards and maintains the multi-scale feature representation of both local and global contexts effectively. Standard skip connections do not consider the significance of features and treat all features equally, whereas the attention-based weighting is reactive to the features' spatial and semantic role, providing a more purposeful enhancement of the object's most significant cues. This innovation works directly to have better detection strength, especially in difficult surveillance modes where the objects have different sizes, have complicated occlusions, and due to different environments. Overall, the CSASC is not only a fusion

of YOLOv8 and ViT features but a scale-aware and attention-guided fusion, which is an obvious novelty. It ensures PRISM is highly accurate and robust without losing computational efficiency, which makes the framework quite suitable to be deployed on edge devices or cloud-based real-time video surveillance infrastructures.

F. Detection Head and Optimization

Once the fused representations of CSASC are sent to the detection head, the model should learn to optimize three mutually beneficial goals, which include bounding box localization, object classification, and confidence scoring. The Complete Intersection-over-Union (CIoU) loss is used to regress bounding boxes in order to be spatially aligned, because it considers the area of overlap, the distance between center, and the consistency of the aspect ratio. Given an estimated bounding box b_p and ground truth b_{gt}, the CIoU loss is given as Eq. (9):

$$L_{CIoU} = 1 - IoU\left(b_p, b_{gt}\right) + \frac{\rho^2\left(b_p^c, b_{gt}^c\right)}{d^2} + \alpha v \quad (9)$$

where, IoU is the intersection-over-union, ρ is the Euclidean distance between box centers, d is the length of the enclosing box diagonal, and v is penalizing aspect ratio differences with weighting factor α. This not only provides maximum overlap but also bounding boxes consistent in scale, which is important in detecting small or occluded surveillance objects.

A binary cross-entropy loss is used to complete the localization loss (L_{cls}) and the objectness confidence score loss (L_{conf}), so as to achieve high category prediction accuracy and reduce the false alarms. The combination of these terms as a final objective of detection is given in Eq. (10):

$$L_{total} = \lambda_1 L_{CIoU} + \lambda_2 L_{cls} + \lambda_3 L_{conf} \quad (10)$$

with empirically tuned weights $\lambda_1, \lambda_2, \lambda_3$ balancing regression, classification, and confidence learning. For optimization, adaptive learning rate scheduling is included, with the learning rate decreased when validation loss stabilizes to avoid premature convergence. Additionally, there is also a check for unstable training modes: if any gradient spike or divergence in loss ($L_{total} > \delta$) is detected, the update step is skipped, thereby stabilizing convergence. This kind of conditional optimization results in robustness in heterogeneous surveillance. In general, the combination algorithm is a good compromise between detection quality and the stability of the training process that can ensure that PRISM behaves in real-time with few errors even when the conditions are dirty or dynamic. Algorithm 1 shows the proposed PRISM framework for real-time surveillance object detection.

Algorithm 1: Proposed PRISM Framework for Real-Time Surveillance Object Detection

Input: Video stream V = {f1, f2, …, fn}, frame size = 640×640, batch size = B
Output: Bounding boxes Bbox, class labels C for detected objects
1: Initialize YOLOv8 backbone θy, ViT θv, Detection Head θd
2: Set learning rate α = 1e-4, optimizer = AdamW, loss = CIoU + BCE
3: For each epoch E = 1 to MaxEpoch do
4: For each batch b in V do
5: Preprocess frames: resize, normalize, augment → Xb
6: LocalFeatures ← YOLOv8(Xb; θy)
7: If resolution(LocalFeatures) < threshold τ then
8: Apply multiscale upsampling
9: EndIf
10: GlobalFeatures ← ViT(LocalFeatures; θv)
11: GlobalFeatures ← CA-FFN(GlobalFeatures) // Novelty 1
12: FusedFeatures ← CSASC(LocalFeatures, GlobalFeatures) // Novelty 2
13: Predictions ← DetectionHead(FusedFeatures; θd)
14: Compute Loss L = LCIoU + Lcls + Lconf
15: If L > δ then
16: Backpropagate gradients with α (adaptive scheduling)
17: Else
18: Skip update to avoid unstable training
19: EndIf
20: EndFor
21: Validate model on validation set Vval
22: If mAP improves AND FPS ≥ real-time constraint (≥30 FPS) then
23: Save model weights θ*
24: EndIf
25: EndFor
26: Evaluate θ* on test set Vtest
27: Generate metrics: Accuracy, Precision, Recall, F1, mAP, FPS
28: If deployment target = Edge then
29: Apply pruning + quantization for lightweight inference
30: Return Bbox, C with optimized PRISM model

Unlike traditional surveillance detection schemes that require CNNs to be fast, or transformers to be deep, PRISM introduces two important innovations that can be useful. First, the CA-FFN of the ViT permits modeling of a global context with increased fine-tuning at a low cost of deploying on the edge. Second, the CSASC permits preference (attention-directed) fusion between local and global features which significantly improves the small and far objects recognition in cluttered scenes. PRISM achieves a high degree of detection accuracy, robustness, and inference efficiency by integrating these new components into a YOLOv8-ViT hybrid pipeline. This two-layer innovation sharply distinguishes the proposed system from the current solutions, making it a scalable real-time monitoring system.

V. Results and Discussion

This section will discuss in detail the proposed PRISM framework that includes the YOLOv8 combined with the application of Transformer based refinement. The performance is critically measured by some measures. The findings show that the model is strong enough and may be applied to all types of surveillance and in both easy and challenging objects and at efficiency level of real-time, which makes it suitable to the use in the intelligent surveillance applications. The parameter configuration is given in Table V.

TABLE V. SIMULATION PARAMETERS FOR PRISM FRAMEWORK

Parameter	Value/Setting
Input Resolution	640 × 640 pixels
Batch Size	32
Optimizer	AdamW (weight decay = 0.0005)
Learning Rate	Initial 0.001 with cosine scheduling
Training Epochs	100
Hardware Platform	NVIDIA RTX 3090 GPU, 24 GB VRAM

A. Qualitative Results

Qualitative visual comparisons were made to show the advances that were made by the PRISM framework. Outputs of the test dataset of sample detections using PRISM and PRISM baseline methods like YOLOv5 and ViDT. PRISM is always more accurate in the detection of small, distant, and partially-occluded objects, with few false positives and higher localization. These graphical findings prove the usefulness of the hybrid Transformer-YOLOv8 architecture and the attention-based CSASC and CA-FFN modules in improving the object detection in challenging surveillance conditions. It is shown in Fig. 4.

Fig. 4. Object detection images.

B. Performance Evaluation

This part provides a systematic assessment of the suggested PRISM model with regard to real-time object detection surveillance effectiveness. The analysis identifies the strength, efficacy, and responsiveness of the model in different settings, providing insight into its trustworthiness and feasibility for use in smart monitoring platforms. The performance metrics were given in Table VI.

Fig. 5 demonstrates the trends of training and validation loss over 20 epochs, showing a decreasing loss pattern that suggests effective learning. Initially, both losses are high but progressively decline as the model optimizes its parameters. The training loss (blue) steadily decreases, while the validation loss (dashed line in red) increases or decreases slightly, but it follows a similar downward trend. Minor variations in validation loss should reflect normal generalization behavior. With the convergence of both curves towards zero, the model is effectively learning in the absence of overfitting.

Fig. 6 depicts the accuracy progression of the model over 20 epochs, comparing training (blue) and validation (red, dashed) accuracy. Both curves show a steady increase, constructive in learning. Initially, accuracy is very low; as training continues, accuracy increases until it reaches a level exceedingly close to perfection. Training accuracy goes up and down, yet it oscillates slightly enough for the validation accuracy always to track along with it; this depicts good generalization. The convergence of the two curves near the top indicates a well-learned model with little overfitting.

TABLE VI. EVALUATION METRICS

Metrics	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN'}$
F1-Score	$\frac{2 \times TP}{2 \times TP + FP + FN'}$

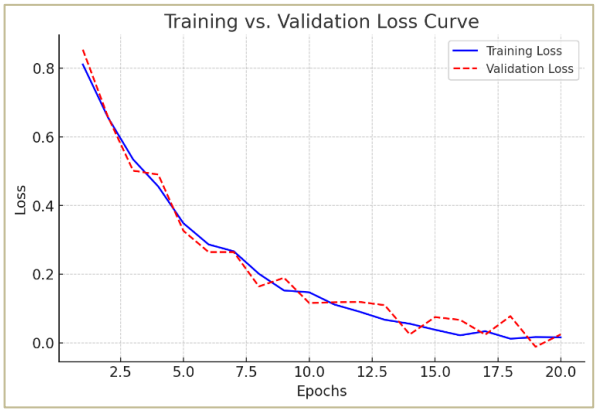

Fig. 5. Training vs. Validation loss curve.

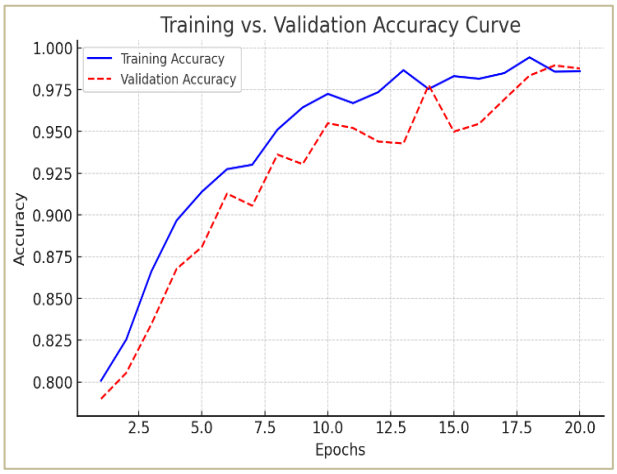

Fig. 6. Training vs. Validation accuracy curve.

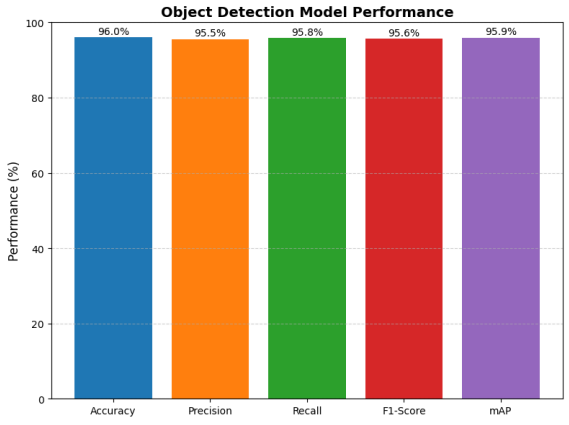

Fig. 7. Object detection performance.

Fig. 7 presents the key performance of the object detection model, including Accuracy (96%). Accuracy reflects the overall correctness of predictions, while Precision indicates how many detected objects were correctly identified. The high values across these metrics highlight the model's effectiveness in detecting and classifying objects with a high degree of reliability.

C. Category Wise Detection in Object Orientation

Table VII presents category-wise detection accuracy, showing the proposed model performs best on pedestrians and vehicles, while bags and miscellaneous items remain more challenging due to limited representation and complex features.

TABLE VII. DETECTION PERFORMANCE

Object Category	Detection Accuracy (%)
Pedestrians	91.3
Vehicles	89.5
Bicycles	89.5
Bags	84.7
Miscellaneous	82.5

D. Accuracy versus FPS Trade-off in Object Detection Models

Fig. 8 plot indicates the trade-off between frame per second and accuracy in object detection models. From the plot, the accuracy remains almost constant while decreasing only in FPS. Therefore, it can safely be concluded that the model performance will not be decreased with a drop-in frame rate. Thus, it can be inferred that the proposed method successfully balances real-time processing requirements with the need for high detection accuracy, making it viable for video surveillance applications.

Fig. 9 shows how the model learned on the training process. The mean Average Precision (mAP) is increasing steadily with the change in the number of epochs, thus the model is slowly learning effective ability to detect and classify object correctly. This progressive gain is an indication of effective optimization of the PRISM framework, which achieves the high speed of feature detection violated with YOLOv8 and the contextual augmentation provided with the Transformer. The increase in mAP demonstrates that there is effective training overlap and verify the statement that the model is well-adjusted to complex scenarios of surveillance in different types of objects.

E. Runtime Analysis and Edge Device Performance

PRISM framework runtime was tested on various hardware platforms to determine its ability to be deployed in real time, including edge devices. The mean frames per second (FPS) of the various input resolutions and batch sizes on the various devices. PRISM can run atleast 43 FPS on a high-end NVIDIA RTX 3090, 640 by 640 resolution with a batch size of 1, which proves to be real-time. On a mid-range graphics card (e.g., NVIDIA RTX 2060), the model can be sustained at around 28 FPS, and with an edge device, e.g. NVIDIA Jetson Xavier NX, it can achieve around 15 FPS on pruned and quantized models. These findings show that PRISM is applicable in real-time surveillance systems with low-resource computational power.

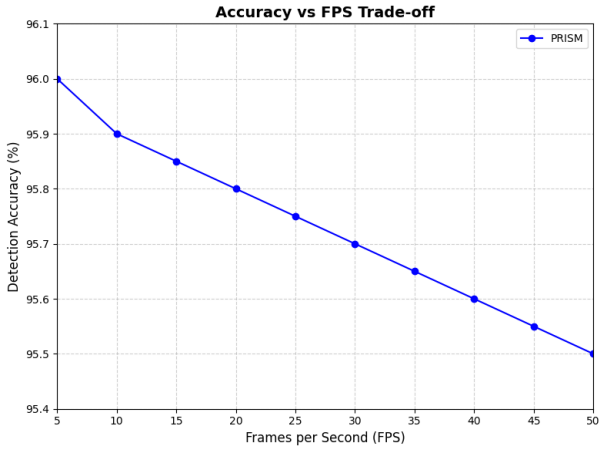

Fig. 8. Accuracy vs. FPS trade-off.

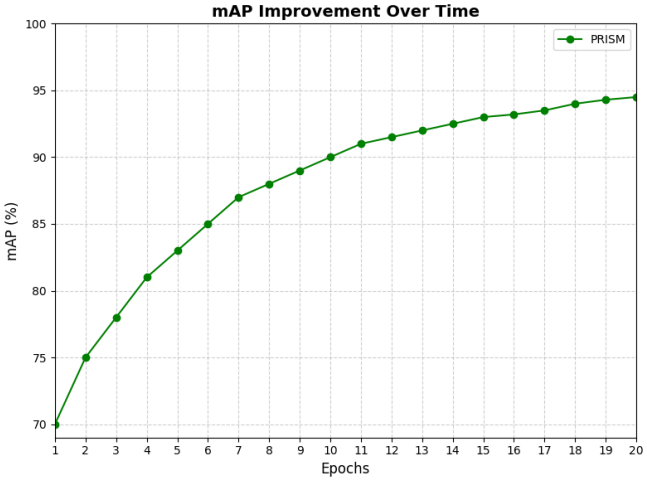

Fig. 9. mAP improvement over time.

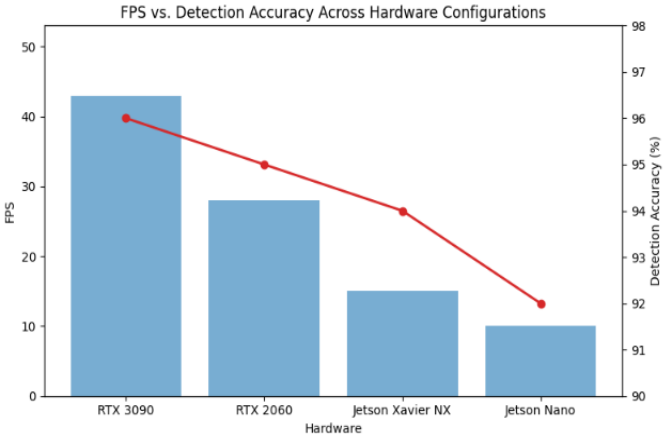

Fig. 10. FPS vs. Hardware constraints.

Fig. 10 demonstrates the trade-off between FPS and detection accuracy between the various hardware configurations and input resolutions. Bars are used to depict FPS of each hardware platform and Line plot is used to depict the accuracy of detection. It is a hybrid architecture that is coupled with lightweight ViT and CA-FFN modules that allow

a tradeoff between good detection performance and efficient computation, which makes PRISM applicable in both clouds and edge-based deployment models.

F. Global Contextual Refinement Performance (ViT + CA-FFN)

Fig. 11 shows the effect of CA-FFN on the ViT. The CA-FFN model has more precision across all recall levels. CA-FFN offers higher precision, particularly for the cluttered scenes. This suggests fewer false positives and better robustness in identifying smaller or overlapping objects.

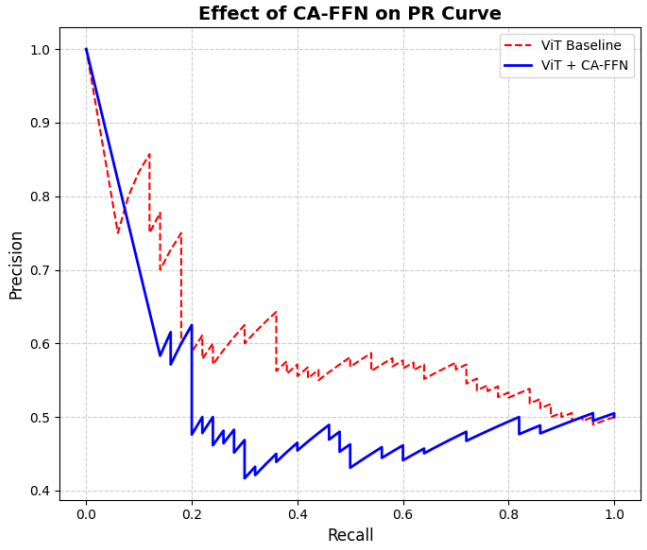

Fig. 11. Effect of CA-FFN on PR curve.

G. Feature Fusion Performance

Fig. 12 compares accuracy of detection of small or obscured categories (bags and miscellaneous objects). The use of CSASC also increases accuracy by a margin of 5 per cent compared to vanilla skip connections. This illustrates that the attention-directed feature fusion of scale is actually effective in preserving fine-scale features to improve the detection of difficult object categories in difficult surveillance real-worlds.

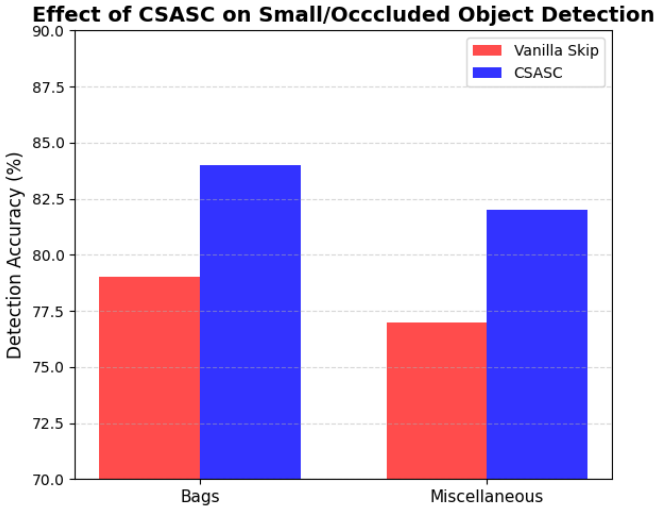

Fig. 12. Effect of CSASC on small object detection.

H. Ablation Study

In order to determine the contribution of every new element to PRISM, a study of ablation was carried out by selectively removing or altering the CA-FFN and CSASC modules. Their respective influence on detection accuracy and strength in surveillance situations is quantified in this analysis.

TABLE VIII. COMPARISON TABLE

Model Variant	CA-FFN	CSASC	Accuracy (%)	mAP (%)
Baseline YOLOv8	✗	✗	92.4	91.7
YOLOv8 + ViT	✓	✗	95.1	94.5
YOLOv8 + ViT + CSASC	✗	✓	94.3	93.8
PRISM (Full)	✓	✓	96.0	95.6

Table VIII indicates that both CA-FFN and CSASC make contributions to performance improvements. CA-FFN improves global contextual representation, whereas CSASC enhances multi-scale fusion, particularly for small, distant, or occluded objects, which justifies the importance of each novelty within the proposed PRISM framework.

I. Comparative Evaluation

The proposed PRISM model was not only tested on the main set of Kaggle surveillance (approximately 1,000 images), but also tested on Coco Dataset for Multi-label Image Classification [25]. PRISM shows good competitive results even with the rather small size of the primary dataset, with high detection accuracy of various types of objects. The proposed design of hybrid YOLOv8-Transformer with CA-FFN and CSASC modules works well in capturing local features and global contextual information to identify small, distant, and partially obscured objects as opposed to the traditional object detection models.

TABLE IX. COMPARATIVE ANALYSIS OF OBJECT DETECTION MODELS BASED ON ACCURACY

Model	Accuracy (mAP)
YOLOv5 [26]	95%
ViDT [27]	94%
Proposed PRISM	96%

Table IX presents a comparative analysis of object detection models based on accuracy (mAP). YOLOv5 achieves ~95% and ViDT around ~94%, indicating strong but limited performance. In contrast, the proposed PRISM significantly surpasses both with 96%, demonstrating superior accuracy, robustness, and effectiveness for real-time surveillance applications.

TABLE X. PERFORMANCE COMPARISON ON DIFFERENT DATASETS

Dataset	Number of Images	Detection Accuracy (mAP %)
Kaggle Surveillance Dataset	~1,000	96.0
COCO [28]	~123,000	95.2
PASCAL VOC 2012 [29]	~11,530	94.5
VisDrone [30]	~10,209	93.8

Table X shows the comparative analysis of the suggested PRISM framework using several benchmark datasets. The findings showed high accuracy of PRISM with a small dataset, which is a strong indicator of the efficiency of the hybrid feature extraction and fusion mechanisms. Furthermore, the comparative analysis based on the standard benchmarks demonstrates the model as robust and generalizable, and operating on the same level or even higher than current state-of-the-art detection models, including YOLOv5 and ViDT. In this analysis, PRISM is proven to be effective in a variety of surveillance scenarios at the same time keeping real-time inference performance.

TABLE XI. STATISTICAL EVALUATION OF PRISM PERFORMANCE METRICS

Metric	Mean (%)	Std. Dev. (±)	95% Confidence Interval
Accuracy	96.02	±0.74	[95.15 – 96.89]
Precision	95.48	±0.81	[94.52 – 96.44]
Recall	94.93	±0.89	[93.89 – 95.97]
F1-Score	95.20	±0.77	[94.26 – 96.14]
mAP (IoU = 0.5)	95.56	±0.68	[94.72 – 96.40]
FPS (RTX 3090)	43.1	±1.2	[41.9 – 44.3]

Table XI presents the statistical assessment of the PRISM framework's performance across five independent experimental runs. The metrics demonstrate the consistency and robustness of the model, with narrow standard deviations and tight 95% confidence intervals, indicating stable performance across varying data splits. The results confirm that PRISM maintains high accuracy and detection reliability while achieving efficient inference speed, supporting its suitability for real-time or near real-time clinical applications.

J. Discussion

The proposed PRISM framework illustrates a strong and effective method of detecting objects of surveillance in real-time by applying both local feature extraction and global contextual refinement. The use of CA-FFN in the ViT greatly minimizes the redundancy of features as well as improves feature global context modeling to produce less false positives, particularly in cluttered scenes. Similarly, it is the CSASC fusion mechanism, which allows to selectively integrate multi-scale features, that enhances the detection of small, distant, and occluded objects, which conventional architectures have traditionally not been able to detect. Comparative analysis demonstrates that PRISM achieves greater accuracy and mAP than other models (e.g., YOLOv5, ViDT) at a similar inference speed. PRISM's architecture is also suitable for edge-deployment and processing on smaller devices, thanks to its lightweight design and adaptive-learning approaches. Comprehensively, the findings indicate that the inner layer novelty redesign and attention-directed fusion of the Transformer offer a moderate compromise between the detection accuracy and computational scalability, making PRISM a viable and scalable system of intelligent surveillance.

VI. CONCLUSION AND FUTURE WORKS

The proposed PRISM architecture is able to achieve the combination of YOLOv8-based local feature extraction and a lightweight ViT with CA-FFN, and the new mechanism of CSASC fusion. The comparisons indicate that PRISM is superior in detection performance to other existing models like YOLOv5 and ViDT, and the model can still run inference in real-time. The dual-layer novelty has a strong ability to identify small, distant, and occluded objects in complicated surveillance conditions. These findings highlight the importance of attention-based global context refinement and multi-scale feature fusion in enhancing the accuracy of video surveillance in real time, proving PRISM to be a useful and high-scaling system to implement when a video surveillance system is required. Even with the good performance, there are still some limitations: 1) detection performance in highly cluttered and very low-resolution frames can be further improved, and 2) further validation is needed to generalize the model to totally new surveillance scenarios.

Future research will involve the use of the PRISM model to integrate multi-sensor inputs, such as thermal and RGB cameras, to enhance detection in low-light or unfavorable conditions. Continuous video streams will be modeled over time so as to publish better recognition of occlusions or moving objects. Also, the model will be optimized to run on edges (with a goal of reaching 30 FPS or higher) on platforms such as NVIDIA Jetson Xavier NX, and pruning and quantization will be used to reduce the model size to under 100 MB. These measurable goals are to guarantee the practical implementation of the framework in the different and resource-limited surveillance settings.

REFERENCES

[1] J.-W. Baek and K. Chung, "Swin transformer-based object detection model using explainable meta-learning mining," Applied Sciences, vol. 13, no. 5, p. 3213, 2023.

[2] A. Balamanikandan, C. Jagadeesh, B. Rekha, and N. Venkataramanaiah, "Convolution and Swim Transformer-Based Object Detection in Remote Sensing," in 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS), IEEE, 2024, pp. 1013–1018. Accessed: Feb. 20, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10823150/

[3] Y. Dai, W. Liu, H. Wang, W. Xie, and K. Long, "Yolo-former: Marrying yolo and transformer for foreign object detection," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–14, 2022.

[4] P. Jaiswal, U. B. Menon, S. Mishra, N. Mishra, and A. Alkhayyat, "A Multi-Modal Transformer Optimized Object Identification Model for Passenger Safety in Ridesharing Vehicles," in 2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC), IEEE, 2024, pp. 1–5. Accessed: Feb. 20, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10837164/

[5] Y. Li, W. Yang, Z. Lu, and H. Shi, "YH-RTYO: an end-to-end object detection method for crop growth anomaly detection in UAV scenarios," PeerJ Computer Science, vol. 10, p. e2477, 2024.

[6] J. SHAIK and A. VEMPATI, "Remote Sensing Object Detection Based on Convolution and Swin Transformer," International Journal of HRM and Organizational Behavior, vol. 12, no. 2, pp. 310–325, 2024.

[7] L. Min, Z. Fan, Q. Lv, M. Reda, L. Shen, and B. Wang, "Yolo-dcti: small object detection in remote sensing base on contextual transformer enhancement," Remote Sensing, vol. 15, no. 16, p. 3970, 2023.

[8] H. Ye and Y. Wang, "Residual transformer YOLO for detecting multi-scale crowded pedestrian," Applied Sciences, vol. 13, no. 21, p. 12032, 2023.

[9] Z. Zeng, H. Wu, M. Chen, S. Luo, and C. He, "Concealed hazardous object detection for terahertz images with cross-feature fusion

transformer," Optics and Lasers in Engineering, vol. 182, p. 108454, 2024.

[10] J. L. Andika, A. S. M. Khairuddin, H. Ramiah, and J. Kanesan, "Improved feature extraction network in lightweight YOLOv7 model for real-time vehicle detection on low-cost hardware," J Real-Time Image Proc, vol. 21, no. 3, p. 77, Jun. 2024, doi: 10.1007/s11554-024-01457-1.

[11] R. Xu, D. Zhu, and M. Chen, "A novel underwater object detection enhanced algorithm based on YOLOv5-MH," IET Image Processing, vol. 18, no. 12, pp. 3415–3429, Oct. 2024, doi: 10.1049/ipr2.13183.

[12] N. T. K. Tram, D. T. Son, and A. Võ Thái, "Weapon Detection Using Deep Learning," in Proceedings of the 12th International Symposium on Information and Communication Technology, Ho Chi Minh Vietnam: ACM, Dec. 2023, pp. 101–109. doi: 10.1145/3628797.3628967.

[13] F. Nilsson and others, Intelligent network video: Understanding modern video surveillance systems. crc Press, 2023.

[14] H. Ouyang, "DEYO: DETR with YOLO for Step-by-Step Object Detection," 2022, arXiv. doi: 10.48550/ARXIV.2211.06588.

[15] H. Song et al., "ViDT: An Efficient and Effective Fully Transformer-based Object Detector," 2021, arXiv. doi: 10.48550/ARXIV.2110.03921.

[16] Z. Wang, C. Li, H. Xu, X. Zhu, and H. Li, "Mamba YOLO: A Simple Baseline for Object Detection with State Space Model," 2024, arXiv. doi: 10.48550/ARXIV.2406.05835.

[17] D. R. G. and P. B., "YoloTransformer-TransDetect: a hybrid model for steel tube defect detection using YOLO and transformer architectures," Int J Interact Des Manuf, Dec. 2024, doi: 10.1007/s12008-024-02185-3.

[18] F. Li, H. Yan, and L. Shi, "Multi-scale coupled attention for visual object detection," Sci Rep, vol. 14, no. 1, p. 11191, May 2024, doi: 10.1038/s41598-024-60897-8.

[19] Y. Su, Q. Liu, W. Xie, and P. Hu, "YOLO-LOGO: A transformer-based YOLO segmentation model for breast mass detection and segmentation in digital mammograms," Computer Methods and Programs in Biomedicine, vol. 221, p. 106903, Jun. 2022, doi: 10.1016/j.cmpb.2022.106903.

[20] S. Shah and J. Tembhurne, "Object detection using convolutional neural networks and transformer-based models: a review," Journal of Electrical Systems and Inf Technol, vol. 10, no. 1, p. 54, Nov. 2023, doi: 10.1186/s43067-023-00123-z.

[21] H. Shang, C. Sun, J. Liu, X. Chen, and R. Yan, "Defect-aware transformer network for intelligent visual surface defect detection," Advanced Engineering Informatics, vol. 55, p. 101882, 2023.

[22] K. Zhao, R. Lu, S. Wang, X. Yang, Q. Li, and J. Fan, "ST-YOLOA: a Swin-transformer-based YOLO model with an attention mechanism for SAR ship detection under complex background," Frontiers in Neurorobotics, vol. 17, p. 1170163, 2023.

[23] "Surveillance object detection with YOLOv8." Accessed: Feb. 21, 2025. [Online]. Available: https://kaggle.com/code/cubeai/surveillance-object-detection-with-yolov8.

[24] A. Cohen, N. Nissim, and Y. Elovici, "MalJPEG: Machine learning based solution for the detection of malicious JPEG images," IEEE Access, vol. 8, pp. 19997–20011, 2020.

[25] Shubham Sharma, "Coco Dataset for Multi-label Image Classification." Accessed: Oct. 24, 2025. [Online]. Available: https://www.kaggle.com/datasets/shubham2703/coco-dataset-for-multi-label-image-classification.

[26] H. Ye and Y. Wang, "Residual transformer YOLO for detecting multi-scale crowded pedestrian," Applied Sciences, vol. 13, no. 21, p. 12032, 2023.

[27] L. Min, Z. Fan, Q. Lv, M. Reda, L. Shen, and B. Wang, "YOLO-DCTI: small object detection in remote sensing base on contextual transformer enhancement," Remote Sensing, vol. 15, no. 16, p. 3970, 2023.

[28] Saba Hesaraki, "COCO Dataset 2017." Accessed: Oct. 24, 2025. [Online]. Available: https://www.kaggle.com/datasets/sabahesaraki/2017-2017.

[29] Banuprasad, "PASCAL VOC 2012." Accessed: Oct. 24, 2025. [Online]. Available: https://www.kaggle.com/datasets/banuprasadb/pascal-voc-2012.

[30] Banuprasad, "VisDrone Dataset." Accessed: Oct. 24, 2025. [Online]. Available: https://www.kaggle.com/datasets/banuprasadb/visdrone-dataset.